AMENDMENTS TO THE CLAIMS

Cancel claims 3, 20, 36 and 44 without prejudice.

This listing of claims will replace all prior versions, and listings, of claims in the application:

1-3. (canceled)

4. (currently amended) ~~The method of claim 3~~

A method in a computer system for dispatching requests to perform services to sub-applications that use different logic models the method comprising:

providing a context for the sub-applications

receiving a request from a client computer to perform a service; and

for a plurality of sub-applications,

determining whether the received request should be dispatched to the sub-application; and

when it is determined that the request should be dispatched to the sub-application, invoking a service routine of the sub-application passing the request

whereby the sub-applications share the provided context;

wherein the determining includes determining whether a match criteria for the sub-application matches the received request;

wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL.

5. (currently amended) The method of claim ~~3~~ 4 including suppressing the invoking of additional service routines when an invoked service routine returns an indication to suppress the invoking of additional service routines.

6. (currently amended) The method of claim ~~3~~ 4 including suppressing the invoking of additional service routines when an invoked service routine responds to the received request.

7. (currently amended) The method of claim ~~3~~ 4 wherein an invoked service routine performs user authentication and indicates to suppress invoking of additional service routines when a user cannot be authenticated.

8. (currently amended) The method of claim ~~3~~ 4 wherein an invoked service routine logs the received request.

9. (canceled)

10. (currently amended) The method of claim ~~3~~ 4 wherein an invoked service routine transforms the received request from one protocol to another protocol.

11. (currently amended) The method of claim ~~3~~ 4 including:

for each of a plurality of sub-applications,

retrieving initialization parameters for the sub-application;

retrieving an indication of a class for the sub-application; and

instantiating an instance of the class with the retrieved initialization parameters.

12. (currently amended) The method of claim ~~3~~ 4 wherein the determining includes determining whether a match criteria in a configuration file for the sub-application matches the received request.

13. (canceled)

14. (currently amended) The method of claim ~~3~~ 4 wherein a sub-application uses an interaction-based model.

15. (currently amended) The method of claim ~~3~~ 4 wherein a sub-application uses an action-view model.

16. (currently amended) The method of claim ~~3~~ 4 wherein a sub-application uses a workflow-based model.

17. (currently amended) The method of claim ~~3~~ 4 wherein the sub-applications form an overall application and wherein the provided context is an application-level context.

18. (currently amended) The method of claim ~~3~~ 4 wherein the sub-applications form an overall application that is web-based.

19. (currently amended) The method of claim ~~3~~ 4 wherein the request is received from a web-server environment.

20. (canceled)

21. (currently amended) ~~The computer system of claim 20~~

A computer system for dispatching HTTP requests to sub-applications, comprising:

a configuration file having a class, initialization parameters, and a match criteria associated with the sub-applications;

an initialization component that instantiates an object of the class for each sub-application in the configuration file, the instantiated object being initialized with the initialization

parameters for the sub-application and being provided with a context object, the context object being shared by the instantiated objects so that the sub-applications share a common context; and

a dispatcher that receives HTTP requests from client computers and, when the received HTTP request matches a match criteria of a sub-application, invokes a service routine of the instantiated object of the class associated with the sub-application;

wherein the match criteria is a regular expression relating to a URL of the HTTP request.

22. (canceled)

23. (currently amended) The computer system of claim ~~20~~ 21 wherein the dispatcher does not invoke any additional service routines when an invoked service routine returns an indication to not invoke any additional service routines.

24. (currently amended) The computer system of claim ~~20~~ 21 wherein the dispatcher does not invoke any additional service routines when an invoked service routine responds to the received request.

25. (currently amended) The computer system of claim ~~20~~ 21 wherein a sub-application is based on an interaction model.

26. (currently amended) The computer system of claim ~~20~~ 21 wherein a sub-application is based on an action-view model.

27. (currently amended) The computer system of claim ~~20~~ 21 wherein each of the sub-applications implement the same interface.

28. (currently amended) A computer system for processing request messages, comprising:

a plurality of sub-applications forming an application, a sub-application having a match criteria indicating when the sub-application should process a request and having a service routine to invoke when the match criteria indicates that the sub-application should process the request, the sub-applications using disparate logic models;

a context for the application that is shared by the sub-applications; and

a dispatcher that receives requests from client computers, evaluates the match criteria to identify which sub-applications have match criteria that match the requests, and invokes the service routines of the identified sub-applications wherein invoked sub-applications use the context;

wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL.

29. (original) The computer system of claim 28 including an initialization component that instantiates an object of a specified class for each sub-application.

30. (original) The computer system of claim 29 wherein the initialization component accesses configuration information that specifies the class of each sub-application and any initialization parameters for the sub-applications.

31. (original) The computer system of claim 29 including a context object representing the context and wherein the initialization component provides the context object to each sub-application.

32. (original) The computer system of claim 28 wherein each service routine is passed a request parameter and returns a response parameter.

33-36. (canceled)

37. (currently amended) A computer system for processing request messages, comprising:

a plurality of service means for servicing requests, the service means forming an application, each service means having a match criteria indicating when the service means should be invoked, the service means implementing different logic models; and

dispatch means for receiving requests from client computers, evaluating match criteria to identify which service means have match criteria that match the requests, and invoking the identified service means whereby the service means share a context that is common to the service means of the application;

wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL.

38. (original) The computer system of claim 37 including an initialization means for instantiating an object of a specified class for each service routine.

39. (original) The computer system of claim 38 wherein the initialization means accesses configuration information that specifies the class of each service means and any initialization parameters for the service means.

40. (original) The computer system of claim 37 wherein each service means is passed a request parameter and returns a response parameter.

41-45. (canceled)

46. (currently amended) ~~The computer-readable medium of claim 44~~

A computer-readable medium for controlling a computer system to dispatch requests to perform services to service routines, by a method comprising:

receiving a request from a client computer to perform a service; and

for a plurality of service routines,

retrieving a match criteria for the service routine;

determining whether the received request matches the retrieved match criteria;

when it is determined that the request matches the retrieved match criteria, invoking the service routine for processing of the received request

whereby the service routines form an application and share a common context;

wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL.

47. (currently amended) The computer-readable medium of claim [[44]] 46 including suppressing the invoking of additional service routines when an invoked service routine returns an indication to suppress the invoking of additional service routines.

48. (currently amended) The computer-readable medium of claim [[44]] 46 including suppressing the invoking of additional service routines when an invoked service routine responds to the received request.

49. (currently amended) The method of claim ~~3~~ 4 wherein all of the sub-applications execute on the same server computer.

50. (currently amended) The computer system of claim ~~20~~ 21 wherein all of the sub-applications execute on the same server computer.

51. (previously presented) The computer system of claim 28 wherein all of the sub-applications execute on the same server computer.

52. (previously presented) The computer system of claim 37, wherein all of the service means and the dispatch means are implemented on the same server computer.

53. (currently amended) The computer-readable medium of claim [[44]] 46 wherein all of the service routines are implemented on the same server computer.

54. (currently amended) The method of claim ~~3~~ 4 wherein a respective service routine is invoked for the request with respect to each of at least two of the sub-applications.

55. (currently amended) The computer system of claim ~~20~~ 21 wherein a respective service routine is invoked for at least one of the HTTP requests with respect to each of at least two of the sub-applications.

56. (previously presented) The computer system of claim 28, wherein a respective service routine is invoked for at least one of the requests with respect to each of at least two of the sub-applications.

57. (previously presented) The computer system of claim 37 wherein at least one of the requests is serviced by at least two of the service means.

58. (previously presented) The computer-readable medium of claim 48 wherein at least one of the requests is processed by at least two of the service routines.

59. (currently amended) The method of claim 54 wherein the sub-applications are ordered and the invoking of the service routines of the at least two sub-applications ~~if~~ is performed in the order of the sub-applications.

60. (previously presented) The computer system of claim 55 wherein the configuration file specifies an ordering of the sub-applications and the dispatcher invokes the service routines of the instantiated objects of the classes associated with the at least two sub-applications in the specified order.

61. (previously presented) The computer system of claim 56 wherein the sub-applications are ordered and the dispatcher invokes the service routines of the at least two sub-applications based on the order of the sub-applications.

62. (previously presented) The computer system of claim 61 wherein an invoked service routine indicates that additional service routines should not be invoked to process the received request.

63. (previously presented) The computer system of claim 61 wherein the dispatcher does not invoke additional service routines when an invoked service routine responds to a received request.

64. (previously presented) The computer system of claim 57 wherein the service means are ordered and the dispatch means invokes the at least two service means based on their order.

65. (previously presented) The computer system of claim 64 wherein an invoked service means indicates that additional service means should not be invoked to process the received request.

66. (previously presented) The computer system of claim 64 wherein the dispatch means does not invoke additional service means when an invoked service means responds to a received request.

67. (previously presented) The computer-readable medium of claim 58 wherein the service routines are ordered and the invoking of the at least two service means is performed in the order of the service routines.